IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Inventor: John McMillan et al.

Application No.: 10/716,337

Filed: November 18, 2003

Title: VIRTUAL OS COMPUTING
ENVIRONMENT

Confirmation No.: 7593

Art Unit: 2194

Examiner: Hoang, Phuong N.

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Commissioner:

This Appeal Brief is submitted in support of a Notice of Appeal to the Board of Patent Appeals and Interferences on March 28, 2008, appealing the final decision of the Office dated October 5, 2007, rejecting all pending claims. Appellant timely filed an After Final Amendment on December 5, 2007. The Office responded with an Advisory Action dated January 11, 2008 and a Supplemental Advisory Action dated January 28, 2008, entering the amendments submitted through After Final Amendment but maintaining the rejections. This Appeal Brief is accompanied by the requisite fee set forth in Rule 41.20(b)(2).

I. Real Party in Interest

The real party in interest in the above-identified application is the assignee, Symantec Corporation, a corporation organized and existing under the laws of the state of Delaware, and having a place of business at 20330 Stevens Creek Blvd., Cupertino, California 95014. The assignment is recorded at Reel/Frame Nos. 019826/0110 at the United States Patent and Trademark Office.

II. Related Appeals and Interferences

Neither the Appellant nor the undersigned attorney is aware of other appeals or interferences that would directly affect or be directly affected by, or have a bearing on, the Board's decision in this appeal.

III. Status of Claims

Claims 1, 3-8, and 10-15 are currently pending in the application and all stand finally rejected. Accordingly, Appellant appeals from the final rejection of claims 1, 3-8, and 10-15, which claims are presented in the Claims Appendix.

IV. Status of Amendments

An After Final Amendment was filed by Appellant on December 5, 2007. The Office entered the amendment for purposes of appeal in an Advisory Action dated January 11, 2008. Subsequent to the Advisory Action, the Office issued a Supplemental Advisory Action dated January 28, 2008 for the purpose of correcting an error in the expiration of the time period of reply. The claims as set forth in the Claims Appendix include the claim amendments as made in the After Final Amendment.

V. Summary of Claimed Subject Matter

Appellant's disclosure describes a virtual operating system (OS) computing machine and methods. As stated in the specification at page 4, lines 1 – 20, the disclosure is directed to the creation of multiple virtual OS environments within a single OS (Figures 1-2). A change made in one environment does not affect the main OS or other environment (Figure 4). As such, each virtual OS environment may appear to an end user to be an independent OS for the applications running within it (Figure 2). The file system and registry information for each environment is independent of the base OS and other environments (Figure 2). Each of the environments may contain a group of installed applications that will run independently of each other (Figures 5-7). The applications running within the virtual OS environments still share the base OS attributes (e.g., networking information, user login rights, services, hardware information, etc.). Also, all of the applications run on a single OS desktop (Figure 1). The end user does not need to be aware that the applications are being run from different virtual OS environments. While the applications share the base OS attributes, any changes made to configuration information may be made into the virtual OS environment and not into the base OS (Figures 3-7).

With respect to independent claim 1, a computing machine having a computing architecture includes a base operating system (OS) installed to the computing machine (Figure 1). The base OS has a base OS file system and a base OS registry (Figure 2). At least one virtual OS environment resides within the base OS, the virtual OS environment having a virtual OS file system and a virtual OS registry which are independent of the base OS file system and the base OS registry (Figure 2). The computing machine is configured such that attempts to access the base OS file system and the base OS registry by an application running under the virtual OS environment are redirected to the virtual OS file system and the virtual OS registry (Figures 3-7).

With respect to independent claim 8, a method of configuring a computer with a base operating system (OS) includes creating at least one virtual OS environment under the base OS such that each virtual OS environment has a virtual file system and registry which are independent of the base OS' file system and registry (Figure 2). The computer is configured such that attempts to access the base OS file system and registry by at least one application running under the virtual OS environment are redirected to the virtual OS environment file system and registry (Figures 3-7).

VI. Grounds of Rejection to be Reviewed on Appeal

1. Whether claims 1, 3-8, and 10-15 are unpatentable under 35 U.S.C. § 101 as being directed to non-statutory subject matter.

2. Whether claims 1, 3-8, and 10-15 are unpatentable under 35 U.S.C. § 103(a) over Hall et al., "A Virtual Operating System," pages 495-502 (hereinafter, "Hall") in view of U.S. Patent No. 6,141,698 to Krishnan et al. (hereinafter, "Krishnan").

VII. Argument

Issue 1 (§101 Rejection)

Claims 1, 3-8, and 10-15 were rejected under 35 U.S.C. §101 ("Section 101") as being directed to non-statutory subject matter, and in particular, a computing architecture comprising software per se. (Final Office Action, page 3). The Office states that the claims "[do] not comprise any hardware to execute the computing machine claimed in the pre-amble." (Final Office Action, page 7). Appellant respectfully disagrees and requests the rejection of these claims not be sustained.

A. Section 101

Section 101 provides the definition for patentable subject matter. It states:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title. (Emphasis added).

The term "process" is defined under 35 U.S.C. § 100 to mean "process, art, or method …" (Emphasis added).

The Supreme Court has held that Congress chose the expansive language of Section 101 so as to include "anything under the sun that is made by man." (Diamond v. Chakrabarty, 447 U.S. 303, 308-09 (1980)). The Court stated that "Congress plainly contemplated that the patent laws would be given wide scope." (Chakrabarty, 447 U.S. at 308-09).

The Federal Circuit has further clarified the expansive scope of Section 101 in stating: "it is improper to read into Section 101 limitations as to the subject matter that may be patented where the legislative history does not indicate that Congress clearly intended such limitations." (In re Alappat, 33 F.3d 1526, 1542 (Fed. Cir. 1994)).

B. Claims 1 and 3-7

Independent claim 1, and claims 3-7 dependent thereon, is directed to:

> 1. A computing machine having a computing architecture, comprising:
>
> a base operating system (OS) installed to the computing machine, the base OS having a base OS file system and a base OS registry;
>
> at least one virtual OS environment within the base OS, the virtual OS environment having a virtual OS file system and a virtual OS registry which are independent of the base OS file system and the base OS registry;
>
> wherein the computing machine is configured such that attempts to access the base OS file system and the base OS registry by an application running under the virtual OS environment are redirected to the virtual OS file system and the virtual OS registry.

(Emphasis added).

First, in addressing the Office's contention, Appellant submits that claims 1 and 3-7 do indeed include "hardware to execute the computing machine claimed in the pre-amble." Claim 1 recites, *inter alia*, that the "computing machine [has] a computing architecture," "a base operating system [is] installed to the computing machine" and "the computing machine is configured such that attempts to access the base OS file system and the base OS registry … are redirected to the virtual OS file system and the virtual OS registry."

Second, claims 1 and 3-7 are directed to a machine that has utility and, as such, meets the statutory requirements of Section 101. The computing machine as claimed may provide certain benefits such as those described at pages 2-3 of Appellant's specification:

> … the creation of multiple, semi-independent virtual OS environments within a single operating system (OS). A change made in one environment does not affect the main OS or any other environment. In this way each virtual OS environment appears to be an independent OS for the applications running within it. The file system and registry information for each environment is independent of the base OS and other environments. Each of the environments can contain a group of installed applications that will run independently of each other.

Further, at page 7 of Appellant's specification, it sets forth:

> Since all of the virtual OS environment information is isolated inside of the base OS file system and registry, it becomes easy to save, store, and load entire virtual OS environments. It is therefore convenient to create a virtual OS environment that contains a set of applications configured in a specific way (or a "clean" virtual OS environment that contains no applications yet) and store it in a separate or central location to be used by multiple computers or the same computer at different times.

Accordingly, Appellant respectfully submits that for at least the aforementioned reasons, claims 1 and 3-7 are directed to statutory subject matter under Section 101 and that the rejection of these claims should not be sustained.

2. Claims 8 and 10-15

Independent claim 8, and claims 10-15 dependent thereon, is directed to a method of configuring a computer. Claim 8 recites:

> 8. A method of configuring a computer with a base operating system (OS) having a base OS file system and registry, the method comprising the steps of:
>
> creating at least one virtual OS environment under the base OS, each virtual OS environment having a virtual file system and registry which are independent of the base OS file system and registry;
>
> configuring the computer such that attempts to access the base OS file system and registry by at least one application running under the virtual OS environment are redirected to the virtual OS environment file system and registry.

(Emphasis added).

In contrast to the Office's contention, Appellant submits that claims 8 and 10-15 do indeed include "hardware to execute the computing machine claimed in the pre-amble." Claim 8 recites, *inter alia*, a "method of configuring a computer" and "configuring the computer such that attempts to access the base OS file system and registry by at least one application running under the virtual OS environment are redirected to the virtual OS environment file system and registry."

Furthermore, Appellant respectfully submits that the method, as claimed in claims 8 and 10-15, is a process that falls under the statutory classes set forth in Section 101. Furthermore, the method provides useful results such as those previously described above.

Accordingly, Appellant respectfully submits that for at least the aforementioned reasons, claims 8 and 10-15 are directed to statutory subject matter under Section 101 and that the rejection of these claims should not be sustained.

Issue 2 (§103 Rejection)

The Office finally rejected claims 1, 3, 5, 6 and 8 under 35 U.S.C. § 102(b) as being anticipated by Hall. (Final Office Action, page 3). Claims 4, 7, and 9-15 were finally rejected under 35 U.S.C § 103(a) ("Section 103") as being unpatentable over Hall in view of Krishnan. (Final Office Action, page 5). An After Final Amendment was filed by Appellant on December 5, 2007 and entered by the Office for purposes of appeal. (Advisory Action dated January 11, 2008, page 1). In the After Final Amendment, certain features recited in dependent claim 9 were incorporated into independent claims 1 and 8 and claim 9 was canceled. As such, claims 1, 3-8 and 10-15 now stand rejected under 35 U.S.C. § 103(a) as being obvious over Hall in view of Krishnan. (Advisory Action, pages 1-2). Appellant respectfully disagrees with this rejection and seek reversal of same.

A. Claims 1 and 8

For at least the reasons discussed below, Hall and Krishnan, taken either alone or in combination, are deficient in teaching or disclosing all of the claim elements recited in independent claims 1 and 8. Moreover, these citations teach away from the invention as claimed.

1. "[A]ttempts to access the base OS file system and the base OS registry by an application running under the virtual OS environment are redirected to the virtual OS file system and the virtual OS registry"

Hall and Krishnan fail to teach or disclose "attempts to access the base OS file system and the base OS registry by an application running under the virtual OS environment are redirected to the virtual OS file system and the virtual OS registry" as recited in claim 1 and "attempts to access the base OS file system and registry by at least one application running under the virtual OS environment are redirected to the virtual OS environment file system and registry" as recited in claim 8. (Emphasis added.)

The Office concedes that the primary citation to Hall does not disclose the aforementioned feature of independent claim 1. (Final Office Action, pages 5, 8). Nonetheless, the Office rejects independent claims 1 and 8, contending that the secondary citation to Krishnan provides the necessary disclosure.

In rejecting original dependent claim 9 (which elements are now affirmatively recited in independent claims 1 and 8), the Office relies on Krishnan and contends:

Krishman [sic] teaches installing at least one application program under the virtual

> OS environment; and wherein attempts to access the base OS file system and registry locations are instead redirected to the virtual OS environment file system or registry.
>
> (Final Office Action, page 6).

The Office points to the Abstract and Figures 2, 3, and 6 of Krishnan where it teaches injecting DLLs. (Final Office Action, page 6). The Office goes on to make the following conclusory statement at page 9:

> It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teaching of Hall and Krishma's [sic] system because Krishma's [sic] injecting dll to the virtual OS would change the execution, but no need to modify the code of the operating system, and therefore, it the maintenance of the application is easier."

Appellant respectfully disagrees with the conclusions of the Office. Appellant finds no evidence or suggestion of the aforementioned claimed features in Krishnan. Further, there is no suggestion to even combine the teachings, as advanced by the Office, except from using Appellant's invention as a template through a hindsight reconstruction of Appellant's claims.

Turning to the Abstract of Krishnan, it teaches:

> [M]ethod and system for modifying the behavior of existing executable code by injecting new code into an executable file. The injection mechanism injects a reference to new code contained in a DLL into an existing executable file such that, when the code of the executable file is executed, the DLL is automatically loaded and the new code is automatically executed. A reference to the DLL is injected into the executable file by either modifying an import table of the file, which causes automatic loading of the DLLs referred to therein, or by adding DLL loader code to the file.
>
> (Krishnan, Abstract.)

Krishnan teaches that injecting code into an executable file may be helpful in situations such as when a third party vendor, who does not have access to application source code, may wish to incorporate vendor-specific code into the application before redistributing to an end customer. (Krishnan, col. 1, lines 23-30.) As another example, Krishnan teaches that injecting new code into existing application code is helpful in preventing software pirates from unencrypting application programs and making illegal copies of such application programs. (Krishnan, col. 1, line 31 – col. 2, line 28.)

However, a simple modification of existing application code using a DLL as disclosed in Krishnan is entirely different from the recitations in Appellant's claims 1 and 8. Clearly, Krishnan does not teach or even contemplate "attempts to access the base OS file system and the base OS registry by an application running under the virtual OS environment are redirected to the virtual OS file system and the virtual OS registry" as recited in claim 1 and "attempts to access the base OS file system and registry by at least one application running under the virtual OS environment are redirected to the virtual OS environment file system and registry" as recited in claim 8.

Appellant submits that Hall does not add anything that would remedy the aforementioned deficiency in Krishnan. Accordingly, for at least this reason, the rejection of independent claims 1 and 8 should not be sustained.

2. "[T]he virtual OS environment having a virtual OS file system and a virtual OS registry which are independent of the base OS file system and the base OS registry"

Hall is an old and dated citation that is alleged to have been published in 1980. In this reference, Hall attempts to solve a problem that existed back in this era, namely, organizations hesitant to move to new hardware systems (with their associated operating systems) because of the high cost of having to train personnel on the new systems and porting software over to the new operating systems. (Hall, page 495, cols. 1-3). Hall proposes a "uniform system interface" that may be run on top of various operating systems and called this interface a "virtual operating system." (Hall, page 495, col. 1 and col. 3.) According to Hall, this is effective because with computer users "there is no need to distinguish between the interface to an operating system and the operating system itself." (Hall, page 495, col. 3.) Fig. 1 of Hall further shows how this virtual machine (i.e. system interface) consists of "interfacing the standardized virtual machine to the vendor supplied system." In essence, this virtual machine (i.e. system interface) is simply a wrapper on top of a base operating system.

Hall does not disclose "the virtual OS environment having a virtual OS file system and a virtual OS registry which are independent of the base OS file system and the base OS registry" as is affirmatively recited in claim 1 or "each virtual OS environment having a virtual file system and registry which are independent of the base OS file system and registry" as is recited in claim 8.

In rejecting these claim elements, the Office alleges that Hall, at page 497, col. 1, section 4, discloses "file systems" and then presumes that a "registry" is inherent for operating systems. (Final Office Action, page 3.) However, even if these assertions are assumed to be accurate for sake of argument, a mere disclosure of "file systems" in Hall and a registry being inherent for an OS, taken alone or together, do not amount to a disclosure of "the virtual OS environment having a virtual OS file system and a virtual OS registry which are <u>independent of the base OS file system and the base OS registry</u>" as is recited in claim 1 or "each virtual OS environment having a virtual file system and registry which are <u>independent of the base OS file system and registry</u>" as is recited in claim 8. (Emphasis added.)

The section of Hall cited by the Office states:

> To test the approach, a uniform program development environment was installed on several distinct systems. A program development environment consists of resources which assist programmers in the development and maintenance of computer programs, such as text editors, programming language processors, and file systems. The types of system resources with which such a virtual machine is concerned (files, directories, processes, and the user environment) require a general-purpose operating system interface.

(Hall, page 497, col. 1, section 4.)

The reference to "file systems" in this section of Hall does not include any teaching or suggestion that the "file systems" are part of a virtual OS environment or that the file systems are independent of a base OS file system. Rather, the reference merely describes resources (e.g., a text editor) that are available in a program development environment to assist programmers in the development and maintenance of computer programs. Moreover, the program development environment in the cited section of Hall is used merely to create source code for a virtual operating system, and there is no teaching or suggestion in Hall of the virtual operation system having a file system that is independent of the base OS file system.

The Office also cites to page 497 of Hall (Final Office Action, page 7):

> In all cases, the system was offered in parallel with the existing environment, thereby allowing users to experiment with the virtual operating system without giving up the familiar, vendor-supplied environment.

(Emphasis added).

The Office uses this citation in an attempt to show the virtual operating system of Hall as being independent of the base OS file system and the base OS registry. However, Appellant's respectfully disagree with this argument. The reference to "parallel" does not mean that the virtual operating system (i.e. system interface) is "independent of the base OS file system and the base OS registry" as is asserted by the Office. Instead, the term "parallel," means simultaneously or concurrently. In other words, the virtual operating system (i.e. system interface) may be run simultaneously or concurrently with the "familiar, vendor-supplied" interface of the base operating system.

In fact, as discussed above, Hall teaches away from "the virtual OS environment having a virtual OS file system and a virtual OS registry which are independent of the base OS file system and the base OS registry" as is recited in claim 1 or "each virtual OS environment having a virtual file system and registry which are independent of the base OS file system and registry" as is recited in claim 8. (Emphasis added.) For instance, the disclosed purpose of the virtual operating system in Hall is to provide a standard interface to a real operating system. (Hall, page 496, Fig. 1 and col. 1, first and second paragraphs.) In other words, the virtual operating system in Hall is simply a wrapper on top of a real operating system, i.e. an interface between a user interface and a real operating system. Hall states, "the emphasis in building a virtual operating system is on the interface presented to the user." (Hall, page 496, col. 1, fourth paragraph.) As a mere interface on top of a real operating system, there is no need for the virtual operating system of Hall to have a file system or a registry that is independent of the file system of registry of the real operating system. In fact, by focusing on providing a standardized interface to a real operating system, Hall teaches away from such a configuration. (Hall, page 495, col. 3, last paragraph.)

Moreover, even if a registry is assumed, *arguendo*, to be inherent for an operating system as asserted by the Office (Final Office Action, page 7), this does not amount to a teaching, explicit or inherent, of a virtual OS environment having a virtual OS registry that is independent of the base OS registry. The virtual operating system of Hall, as a mere interface to a real operating system, teaches away from an independent registry for the virtual operating system.

For at least these reasons, Hall fails to disclose "the virtual OS environment having a virtual OS file system and a virtual OS registry which are independent of the base OS file system and the base OS registry" as is recited in claim 1 or "each virtual OS environment having a virtual file system and registry which are independent of the base OS file system and registry" as is recited in claim 8. (Emphasis added.)

Krishnan is cited for its alleged disclosure of various features of claim 1 other than the aforementioned feature. Appellant's respectfully submit that Krishnan does not add anything to the disclosure of Hall that would remedy the aforementioned deficiency.

Accordingly, for at least the aforementioned reason, the rejection of independent claims 1 and 8 should be not be sustained.

3. "[A]t least one virtual OS environment within the base OS"

Hall and Krishnan also fail to disclose other elements recited in independent claim 1. For example, Hall and Krishnan fail to disclose "at least one virtual OS environment within the base OS" as recited in claim 1 and "creating at least one virtual OS environment under the base OS" as recited in claim 8. (Emphasis added.)

In fact, Hall and Krishnan teach away from these claim elements. For example, Fig. 1 of Hall shows that the virtual operating system, as an interface to a real operating system, is on top of a vendor supplied system (i.e., a wrapper for a real operating system). Therefore, Hall fails to disclose "at least one virtual OS environment within the base OS" as recited in claim 1 and "creating at least one virtual OS environment under the base OS" as recited in claim 8. (Emphasis added.) For this independent reason, the rejection of independent claims 1 and 8 should not be sustained.

B. Claims 3-7, 10, and 11

As discussed above, neither Hall nor Krishnan, either alone or in combination, teach, disclose or suggest all claim elements recited in claims 1 and 8. Therefore, the Section 103 rejection of claims 3-7, 10 and 11 should not be sustained at least based upon their dependencies from independent claims 1 or 8.

C. Claim 12

The rejection of claim 12 should not be sustained because claim 12 depends from independent claim 8. In addition, claim 12 recites subject matter not disclosed in Hall or Krishnan. For example, claim 12 recites "creating a copy of the base OS file system and registry in the virtual OS environment file system and registry."

The Office asserts that Krishnan teaches this feature at column 4, line 55 – column 5, line 5. (Final Office Action, page 6.) The cited section of Krishnan states:

> The injection mechanism is useful in many scenarios. For example, in a globally networked system such as the Internet, licensing code can be incorporated into an existing application and distributed on the system by injecting the licensing code into the application using the injection mechanism. The licensing developer creates a new DLL with the new licensing code accessible through the initialization function of the DLL. The developer then uses the injection mechanism of the present invention to create a modified version of the application that includes a reference to the new DLL. This modified version is then distributed. Further, the newly injected licensing code can be made more secure by using the injection mechanism to inject security code into the modified application. The injected security code makes it impossible to recreate in a reasonable amount of time an unmodified version of the application that does not include the injected licensing DLL.

Appellant respectfully submits the above citation is silent as to the subject feature. Further, Appellant has carefully reviewed the remainder of Krishnan and Hall and finds no teaching of "creating a copy of the base OS file system and registry in the virtual OS environment file system and registry." Thus, for at least these reasons, the rejection of dependent claim 12 should not be sustained.

D. Claim 13

The rejection of claim 13 should not be sustained because claim 13 depends indirectly from independent claim 8. In addition, claim 13 recites subject matter neither disclosed in Hall nor Krishnan. Claim 13 recites "wherein an application running under the virtual OS environment is executed using the copy in the virtual OS environment file system and registry."

The Office rejects this claim under the same rationale used for claim 12, namely, relying on the cited section of Krishnan disclosed above. (Final Office Action, page 6.)

Again, Appellant respectfully submits that this citation is silent as to the subject feature and that neither Krishnan (nor Hall) teach or suggest "wherein an application running under the virtual OS environment is executed using the copy in the virtual OS environment file system and registry" as is affirmatively recited in claim 13. Accordingly, for at least these reasons, the rejection of dependent claim 13 should not be sustained.

E. Claim 14

The rejection of claim 14 should not be sustained at least because claim 14 depends directly from independent claim 8. In addition, claim 14 recites subject matter not disclosed in Hall and Krishnan. Claim 14 requires "setting a predetermined directory such that an application running under the predetermined directory will be redirected to the virtual OS environment based on the location of the application being under the predetermined directory."

Regarding claim 14, the Office contends:

> [I]t would have been obvious to one of ordinary skill in the art at the time the invention was made to recognize that the injecting DLL method and techniques can be applied to different computing environment because the injecting DLL is portable and convenient to attach anywhere in the application as designed to redirect the execution without being modified the existing code.

(Final Office Action, page 7.)

The Office's argument is unclear as it does not substantively addresses the features recited in claim 14. Appellant has carefully reviewed the cited references, namely Hall and Krishnan, and submits that neither Krishnan nor Hall teach "setting a predetermined directory such that an application running under the predetermined directory will be redirected to the virtual OS environment based on the location of the application being under the predetermined directory." Accordingly, for at least these reasons, the rejection of dependent claim 14 should not be sustained.

F. Claim 15

As discussed above, neither Hall nor Krishnan, either alone or in combination, teach, disclose or suggest all claim elements recited in claim 14. Therefore, the Section 103 rejection of claim 15 should not be sustained at least based upon its dependency from claim 14.

Conclusion

In view of the foregoing, it is submitted that the final rejections of the pending claims are improper and should not be sustained. Therefore, a reversal of the final rejections in the Office Action dated October 5, 2007 is respectfully requested.

It is believed that any fees associated with the filing of this paper are identified in an accompanying transmittal. However, if any additional fees are required, they may be charged to Deposit Account 50-3583, in the name of AdvantEdge Law Group, LLC. To the extent necessary, a petition for extension of time under 37 C.F.R. 1.136(a) is hereby made, the fee for which should be charged against the aforementioned account.

Respectfully submitted,

Date: May 22, 2008

By: /L. Eric Gardanier/
L. Eric Gardanier
Attorney for Appellant
Registration No. 40,617
Customer No. 53124

VIII. Claims Appendix

1. A computing machine having a computing architecture, comprising:

a base operating system (OS) installed to the computing machine, the base OS having a base OS file system and a base OS registry;

at least one virtual OS environment within the base OS, the virtual OS environment having a virtual OS file system and a virtual OS registry which are independent of the base OS file system and the base OS registry;

wherein the computing machine is configured such that attempts to access the base OS file system and the base OS registry by an application running under the virtual OS environment are redirected to the virtual OS file system and the virtual OS registry.

3. The computing machine of claim 1, wherein the application running under the virtual OS environment shares one or more of the following with the base OS:

networking information,
user login rights,
services,
hardware information, and
clipboard information.

4. The computing machine of claim 1, further including multiple virtual OS environments within the base OS, and wherein a change made in one of the virtual OS environments does not affect the base OS or any of the other virtual OS environments.

5. The computing machine of claim 1, wherein each virtual OS environment contains a group of installed applications that run independently of one another.

6. The computing machine of claim 1, further including one or more applications running under the base OS and each virtual OS environment, and wherein all of the applications run on a single OS desktop.

7. The computing machine of claim 1, wherein a change made to configuration

information with respect to the virtual OS environment does not change configuration information associated with the base OS.

8. A method of configuring a computer with a base operating system (OS) having a base OS file system and registry, the method comprising the steps of:

creating at least one virtual OS environment under the base OS, each virtual OS environment having a virtual file system and registry which are independent of the base OS file system and registry;

configuring the computer such that attempts to access the base OS file system and registry by at least one application running under the virtual OS environment are redirected to the virtual OS environment file system and registry.

10. The method of claim 8, further including the step of altering one or more application programming interfaces (APIs) that access the base OS file system and registry directly and indirectly so as to redirect these accesses into the appropriate virtual file system and registry.

11. The method of claim 10, further including the step of injecting a DLL into every application that is executed.

12. The method of claim 8, further including the step of creating a copy of the base OS file system and registry in the virtual OS environment file system and registry.

13. The method of claim 12, wherein the application running under the virtual OS environment is executed using the copy in the virtual OS environment file system and registry.

14. The method of claim 8, further including setting a predetermined directory such that an application running under the predetermined directory will be redirected to the virtual OS environment based on the location of the application being under the predetermined directory.

15. The method of claim 14, wherein the predetermined directory is a CD/DVD drive in the base OS file system.

IX. Evidence Appendix

None.

X. Related Proceedings Appendix

None.